

ANTI-COMMUTATIVE DUAL COMPLEX NUMBERS AND 2D RIGID TRANSFORMATION

GENKI MATSUDA, SHIZUO KAJI[†], AND HIROYUKI OCHIAI

ABSTRACT. We introduce a new presentation of the two dimensional rigid transformation which is more concise and efficient than the standard matrix presentation. By modifying the ordinary dual number construction for the complex numbers, we define the ring of the *anti-commutative dual complex numbers*, which parametrizes two dimensional rotation and translation all together. With this presentation, one can easily interpolate or blend two or more rigid transformations at a low computational cost. We developed a library for C++ with the MIT-licensed source code ([13]) and demonstrate its facility by an interactive deformation tool developed for iPad.

1. RIGID TRANSFORMATION

The n -dimensional *rigid transformation (or Euclidean)* group $E(n)$ consists of transformations of \mathbb{R}^n which preserves the standard metric. This group serves as an essential mathematical backend for many applications (see [2, 9]). It is well-known (see [6], for example) that any element of $E(n)$ can be written as a composition of a rotation, a reflection, and a translation, and hence, it is represented by $(n + 1) \times (n + 1)$ -homogeneous matrix;

$$E(n) = \left\{ A = \begin{pmatrix} \hat{A} & d_A \\ 0 & 1 \end{pmatrix} \mid \hat{A}'\hat{A} = I_n, d_A \in \mathbb{R}^n \right\}.$$

Here, we adopt the convention that a matrix acts on a (column) vector by the multiplication from the left. $E(n)$ has two connected components. The identity component $SE(n)$ consists of those without reflection. More precisely,

$$SE(n) = \left\{ A = \begin{pmatrix} \hat{A} & d_A \\ 0 & 1 \end{pmatrix} \mid \hat{A} \in SO(n), d_A \in \mathbb{R}^n \right\},$$

where $SO(n) = \{R \mid R'R = I_n, \det(R) = 1\}$ is the *special orthogonal group* composed of n -dimensional rotations.

The group $SE(n)$ is widely used in computer graphics such as for expressing motion and attitude, displacement ([10]), deformation ([1, 5, 11]), skinning ([8]), and camera control ([3]). In some cases, the matrix representation of the group $SE(n)$ is not convenient. In particular, a linear combination of two matrices in $SE(n)$ does not necessarily belong to $SE(n)$ and

Key words and phrases. 2D rigid transformation, 2D Euclidean transformation, dual numbers, dual quaternion numbers, dual complex numbers, deformation.

[†]Corresponding author.

it causes the notorious *candy-wrapper* defect in skinning. When $n = 3$, another representation of $SE(3)$ using the *dual quaternion numbers* (DQN, for short) is considered in [8] to solve the candy-wrapper defect. In this paper, we consider the 2-dimensional case. Of course, 2D case can be handled by regarding the plane embedded in \mathbb{R}^3 and using DQN, but instead, we introduce the *anti-commutative dual complex numbers* (DCN, for short), which is specific to 2D with much more concise and faster implementation ([13]). To summarise, our DCN has the following advantages:

- any number of rigid transformations can be blended/interpolated easily using its algebraic structure with no degeneration defects such as the candy-wrapper defect (see §5)
- it is efficient in terms of both memory and CPU usage (see §6).

We believe that our DCN offers a choice for representing the 2D rigid transformation in certain applications which requires the above properties.

2. ANTI-COMMUTATIVE DUAL COMPLEX NUMBERS

Let \mathbb{K} denote one of the fields \mathbb{R}, \mathbb{C} , or \mathbb{H} , where \mathbb{H} is the quaternion numbers. First, we recall the standard construction of the *dual numbers* over \mathbb{K} .

Definition 2.1. The ring of dual numbers $\hat{\mathbb{K}}$ is the quotient ring defined by

$$\hat{\mathbb{K}} := \mathbb{K}[\varepsilon]/(\varepsilon^2) = \{p_0 + p_1\varepsilon \mid p_0, p_1 \in \mathbb{K}\}.$$

We often denote an element in $\hat{\mathbb{K}}$ by a symbol with hat such as \hat{p} .

The addition and the multiplication of two dual numbers are given as

$$\begin{aligned} (p_0 + p_1\varepsilon) + (q_0 + q_1\varepsilon) &= (p_0 + q_0) + (p_1 + q_1)\varepsilon, \\ (p_0 + p_1\varepsilon)(q_0 + q_1\varepsilon) &= (p_0q_0) + (p_1q_0 + p_0q_1)\varepsilon. \end{aligned}$$

The following involution is considered to be the dual version of conjugation

$$p_0 + \widetilde{p_1\varepsilon} := p_0^* - p_1^*\varepsilon,$$

where p_i^* is the usual conjugation of p_i in \mathbb{K} . (Note that in some literatures $\tilde{\hat{p}}$ is denoted by $\overline{\hat{p}^*}$.)

The *unit dual numbers* are of special importance.

Definition 2.2. Let $|\hat{p}| = \sqrt{\hat{p}\tilde{\hat{p}}}$ for $\hat{p} \in \hat{\mathbb{K}}$. The unit dual numbers is defined as

$$\hat{\mathbb{K}}_1 := \{\hat{p} \in \hat{\mathbb{K}} \mid |\hat{p}| = 1\} \subset \hat{\mathbb{K}}.$$

$\hat{\mathbb{K}}_1$ acts on $\hat{\mathbb{K}}$ by conjugation action

$$\hat{p} \diamond \hat{q} := \hat{p}\hat{q}\tilde{\hat{p}}$$

where $\hat{p} \in \hat{\mathbb{K}}_1, \hat{q} \in \hat{\mathbb{K}}$.

The unit dual quaternion $\hat{\mathbb{H}}_1$ is successfully used for skinning in [8]; a vector $v = (x, y, z) \in \mathbb{R}^3$ is identified with $1 + (xi + yj + zk)\varepsilon \in \hat{\mathbb{H}}_1$ and the conjugation action of $\hat{\mathbb{H}}_1$ preserves the embedded \mathbb{R}^3 and its Euclidean metric. In fact, the conjugation action induces the double cover $\hat{\mathbb{H}}_1 \rightarrow \text{SE}(3)$.

On the other hand, when $\mathbb{K} = \mathbb{R}$ or \mathbb{C} , the conjugation action is trivial since the corresponding dual numbers are commutative. Therefore, we define the *anti-commutative dual complex numbers* (DCN, for short) $\check{\mathbb{C}}$ by modifying the multiplication of $\check{\mathbb{C}}$. That is, $\check{\mathbb{C}} = \check{\mathbb{C}}$ as a set, and the algebraic operations are replaced by

$$\begin{aligned} (p_0 + p_1\varepsilon)(q_0 + q_1\varepsilon) &= (p_0q_0) + (p_1\tilde{q}_0 + p_0q_1)\varepsilon, \\ \widetilde{p_0 + p_1\varepsilon} &= \tilde{p}_0 + p_1\varepsilon, \\ |p_0 + p_1\varepsilon| &= |p_0|. \end{aligned}$$

The addition and the conjugation action are kept unchanged. Then,

Theorem 2.3. $\check{\mathbb{C}}$ satisfies distributive and associative laws, and hence, has a (non-commutative) ring structure.

Similarly to the unit dual quaternion numbers, the unit anti-commutative complex numbers are of particular importance:

$$\check{\mathbb{C}}_1 := \{\hat{p} \in \check{\mathbb{C}} \mid |\hat{p}| = 1\} = \{e^{i\theta} + p_1\varepsilon \in \check{\mathbb{C}} \mid \theta \in \mathbb{R}, p_1 \in \mathbb{C}\}.$$

It forms a group with inverse

$$(e^{i\theta} + p_1\varepsilon)^{-1} = e^{-i\theta} - p_1\varepsilon.$$

We define an action of $\check{\mathbb{C}}_1$ on $\check{\mathbb{C}}$ by the conjugation. Now, we regard $\mathbb{C} = \mathbb{R}^2$ as usual. Identifying $v \in \mathbb{C}$ with $1 + v\varepsilon \in \check{\mathbb{C}}$, we see that $\check{\mathbb{C}}_1$ acts on \mathbb{C} as rigid transformation.

3. RELATION TO SE(2)

In the previous section, we constructed the unit anti-commutative dual complex numbers $\check{\mathbb{C}}_1$ and its action on $\mathbb{C} = \mathbb{R}^2$ as rigid transformation. Recall that $\hat{p} = p_0 + p_1\varepsilon \in \check{\mathbb{C}}_1$ acts on $v \in \mathbb{C}$ by

$$(3.1) \quad \hat{p} \diamond (1 + v\varepsilon) = (p_0 + p_1\varepsilon)(1 + v\varepsilon)(\tilde{p}_0 + p_1\varepsilon) = 1 + (p_0^2v + 2p_0p_1)\varepsilon,$$

that is, v maps to $p_0^2v + 2p_0p_1$. For example, when $p_1 = 0$, $v \in \mathbb{C}$ is mapped to p_0^2v , which is the rotation around the origin of degree $2 \arg(p_0)$ since $|p_0| = 1$. On the other hand, when $p_0 = 1$, the action corresponds to the translation by $2p_1$. In general, we have

$$\begin{aligned} \varphi : \check{\mathbb{C}}_1 &\rightarrow \text{SE}(2) \\ p_0 + p_1\varepsilon &\mapsto \begin{pmatrix} \text{Re}(p_0^2) & -\text{Im}(p_0^2) & \text{Re}(2p_0p_1) \\ \text{Im}(p_0^2) & \text{Re}(p_0^2) & \text{Im}(2p_0p_1) \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

where $\operatorname{Re}(2p_0p_1)$ (respectively, $\operatorname{Im}(2p_0p_1)$) is the real (respectively, imaginary) part of $2p_0p_1 \in \mathbb{C}$. Note that this gives a surjective group homomorphism $\varphi : \check{\mathbb{C}}_1 \rightarrow \operatorname{SE}(2)$ whose kernel is $\{\pm 1\}$. That is, the preimage of any 2D rigid transformation consists of exactly two unit DCN's with opposite signs.

Example 3.1. We compute the DCN's $\pm \hat{p} \in \check{\mathbb{C}}_1$ which represent θ -rotation around $v \in \mathbb{C}$. It is the composition of the following three rigid transformations: $(-v)$ -translation, θ -rotation around the origin, and v -translation. Thus, we have

$$\hat{p} = \left(1 + \frac{v}{2}\varepsilon\right) \cdot \left(\pm e^{\frac{\theta}{2}i}\right) \cdot \left(1 - \frac{v}{2}\varepsilon\right) = \pm \left(e^{\frac{\theta}{2}i} + \left(e^{-\frac{\theta}{2}i} - e^{\frac{\theta}{2}i}\right) \frac{v}{2}\varepsilon\right).$$

4. RELATION TO THE DUAL QUATERNION NUMBERS

The following ring homomorphism

$$\check{\mathbb{C}} \ni p_0 + p_1\varepsilon \mapsto p_0 + p_1j\varepsilon \in \hat{\mathbb{H}}$$

is compatible with the involution and the conjugation, and preserves the norm. Furthermore, if we identify $v = (x, y) \in \mathbb{R}^2$ with $1 + (xj + yk)\varepsilon (= 1 + (x + yi)j\varepsilon)$, the above map commutes with the action. From this point of view, DCN is nothing but a sub-ring of DQN.

Note also that $\check{\mathbb{C}}$ can be embedded in the ring of the 2×2 -complex matrices by

$$p_0 + p_1\varepsilon \mapsto \begin{pmatrix} p_0 & p_1 \\ 0 & \tilde{p}_0 \end{pmatrix}.$$

Then

$$\widetilde{\begin{pmatrix} p_0 & p_1 \\ 0 & \tilde{p}_0 \end{pmatrix}} = \begin{pmatrix} \tilde{p}_0 & p_1 \\ 0 & p_0 \end{pmatrix}, \quad \left| \begin{pmatrix} p_0 & p_1 \\ 0 & \tilde{p}_0 \end{pmatrix} \right|^2 = \det \begin{pmatrix} p_0 & p_1 \\ 0 & \tilde{p}_0 \end{pmatrix}.$$

We thus have various equivalent presentations of DCN. However, our presentation of $\check{\mathbb{C}}$ as the anti-commutative dual numbers is easier to implement and more efficient.

5. INTERPOLATION OF 2D RIGID TRANSFORMATIONS

First, recall that for the positive real numbers $x, y \in \mathbb{R}_{>0}$, there are two typical interpolation methods:

$$(1 - t)x + ty, \quad t \in \mathbb{R}$$

and

$$(yx^{-1})^t x, \quad t \in \mathbb{R}.$$

The first method can be generalized to DQN as the *Dual quaternion Linear Blending* in [8]. Similarly, a DCN version of *Dual number Linear Blending* (*DLB, for short*) is given as follows:

Definition 5.1. For $\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n \in \check{\mathbb{C}}_1$, we define

$$(5.1) \quad DLB(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n; w_1, w_2, \dots, w_n) = \frac{w_1 \hat{p}_1 + w_2 \hat{p}_2 + \dots + w_n \hat{p}_n}{|w_1 \hat{p}_1 + w_2 \hat{p}_2 + \dots + w_n \hat{p}_n|},$$

where $w_1, w_2, \dots, w_n \in \mathbb{R}$. Note that the denominator can become 0 and for those set of w_i 's and \hat{p}_i 's DLB cannot be defined.

A significant feature of DLB is that it is distributive (it is called *bi-invariant* in some literatures). That is, the following holds:

$$\hat{p}_0 DLB(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n; w_1, w_2, \dots, w_n) = DLB(\hat{p}_0 \hat{p}_1, \hat{p}_0 \hat{p}_2, \dots, \hat{p}_0 \hat{p}_n; w_1, w_2, \dots, w_n).$$

This property is particularly important when transformations are given in a certain hierarchy such as in the case of skinning; if the transformation assigned to the root joint is modified, the skin associated to lower nodes is deformed consistently.

Next, we consider the interpolation of the second type. For this, we need the exponential and the logarithm maps for DCN.

Definition 5.2. For $\hat{p} = p_0 + p_1 \varepsilon \in \check{\mathbb{C}}$, we define

$$\exp \hat{p} = \sum_{n=0}^{\infty} \frac{(p_0 + p_1 \varepsilon)^n}{n!} = e^{p_0} + \frac{(e^{p_0} - e^{\tilde{p}_0})}{p_0 - \tilde{p}_0} p_1 \varepsilon.$$

When $\exp \hat{p} \in \check{\mathbb{C}}_1$, we can write $p_0 = \theta i$ for some $-\pi \leq \theta < \pi$, and

$$\exp \hat{p} = e^{\theta i} + \frac{\sin \theta}{\theta} p_1 \varepsilon.$$

For $\hat{q} = e^{\theta i} + q_1 \varepsilon \in \check{\mathbb{C}}_1$, we define

$$\log(\hat{q}) = \theta i + \frac{\theta}{\sin \theta} q_1 \varepsilon.$$

As usual, we have $\exp(\log(\hat{q})) = \hat{q}$ and $\log(\exp(\hat{p})) = \hat{p}$. Note that this gives the following *Lie correspondence* (see [4, 2, 9])

$$\exp : \mathfrak{dcn} \rightarrow \check{\mathbb{C}}_1,$$

$$\log : \check{\mathbb{C}}_1 \rightarrow \mathfrak{dcn},$$

where $\mathfrak{dcn} = \{\theta i + p_1 \varepsilon \in \check{\mathbb{C}} \mid \theta \in \mathbb{R}, p_1 \in \mathbb{C}\} \simeq \mathbb{R}^3$. We have the following commutative diagram:

$$\begin{array}{ccc} \mathfrak{dcn} & \xrightarrow{d\varphi} & \mathfrak{se}(2) \\ \downarrow \exp & & \downarrow \exp \\ \check{\mathbb{C}}_1 & \xrightarrow{\varphi} & \mathrm{SE}(2), \end{array}$$

where

$$d\varphi : \mathfrak{dcn} \rightarrow \mathfrak{se}(2)$$

$$\theta i + (x + yi)\varepsilon \mapsto \begin{pmatrix} 0 & -2\theta & 2x \\ 2\theta & 0 & 2y \\ 0 & 0 & 0 \end{pmatrix},$$

is an isomorphism of \mathbb{R} -vector spaces.

The following is a DCN version of SLERP [12].

Definition 5.3. SLERP interpolation from $\hat{p} \in \check{\mathbb{C}}_1$ to $\hat{q} \in \check{\mathbb{C}}_1$ is given by

$$\text{SLERP}(\hat{p}, \hat{q}; t) = (\hat{q}\hat{p}^{-1})^t \hat{p} = \exp(t \log(\hat{q}\hat{p}^{-1}))\hat{p},$$

where $t \in \mathbb{R}$.

This gives a uniform angular velocity interpolation of two DCN's, while DLB can blend three or more DCN's without the uniform angular velocity property.

6. COMPUTATIONAL COST

We compare the following four methods for 2D rigid transformation: our DCN, the unit dual quaternion numbers (see §4), the 3×3 -real (homogeneous) matrix representation of $\text{SE}(2)$, and the 2×2 -complex matrices described below.

We list the computational cost in terms of the number of floating point operations for

- transforming a point
- composing two transformations
- converting a transformation to the standard 3×3 -real matrix representation (except for the 3×3 -real matrix case where it shows the computational cost to convert to DCN representation).

We also give the memory usage for each presentation in terms of the number of floating point units necessary to store a transformation.

	transformation	composition	conversion	memory usage
DCN	22 FLOPs	20 FLOPs	15 FLOPs	4 scalars
DQN	92 FLOPs	88 FLOPs	NA	8 scalars
2×2 -complex matrix	112 FLOPs	56 FLOPs	15 FLOPs	8 scalars
3×3 -real matrix	15 FLOPs	45 FLOPs	18 FLOPs(to DCN)	9 scalars

TABLE 1. Comparison of computational cost

Note that when a particular application requires to apply a single transformation to a lot of points, it is faster to first convert the DCN to a 3×3 -real matrix.

7. A C++ LIBRARY

We implemented our DCN in a form of C++ library. Though it is written in C++, it should be easy to translate to any language. You can download the MIT-licensed source code at [13]. We also developed a small demo application called the 2D probe-based deformer ([7]) for tablet devices running OpenGL ES (OpenGL for Embedded Systems). Thanks to the efficiency of DCN and touch interface, it offers interactive and intuitive

operation. Although we did not try, we believe DCN works well with 2D skinning just as DQN does with 3D skinning.

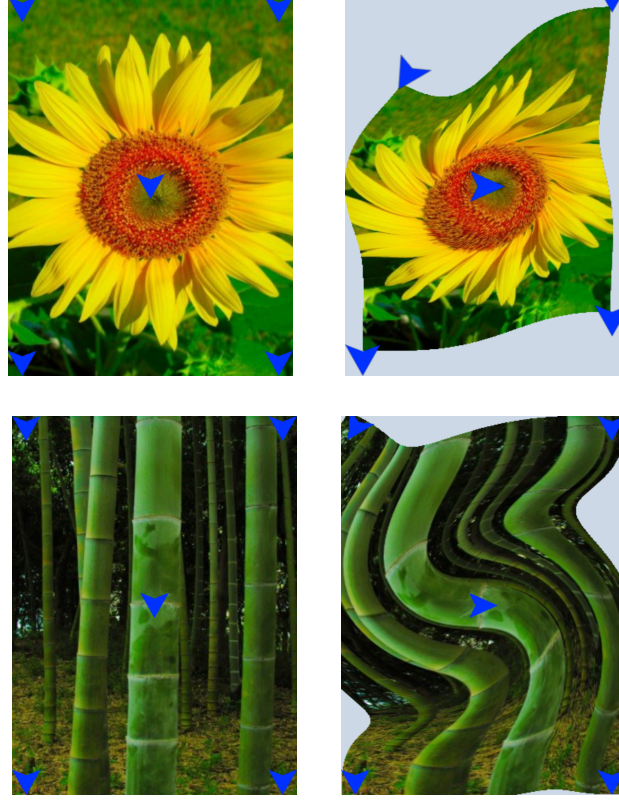


FIGURE 1. Left: set up initial positions of (blue) probes. Right: the target picture is deformed according to user's action on the probes.

Here we briefly discuss the algorithm of the application. One can place an arbitrary number of *probes* P_1, \dots, P_n on a target image. The target image is represented by a textured square mesh with vertices v_1, \dots, v_m . The weight w_{ij} of P_i 's effect on v_j is painted by user or automatically calculated from the distance between v_j and P_i . When the probes are rotated or translated by the user's touch gesture, the DCN \hat{p}_i is computed which maps the initial state of P_i to its current state. The vertex v_j is transformed by

$$v_j \mapsto DLB(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n; w_{1j}, w_{2j}, \dots, w_{nj}) \diamond v_j.$$

(See Equations (3.1) and (5.1) respectively for the definition of the action \diamond and DLB.)

ACKNOWLEDGEMENTS

This work was supported by Core Research for Evolutional Science and Technology (CREST) Program "Mathematics for Computer Graphics" of Japan Science and Technology Agency (JST). The authors are grateful for S.

Hirose at OLM Digital Inc., and Y. Mizoguchi, S. Yokoyama, H. Hamada, and K. Matsushita at Kyushu University for their valuable comments.

REFERENCES

- [1] M. Alexa, D. Cohen-Or, and D. Levin, *As-rigid-as-possible shape interpolation*, In Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 157–164. DOI=10.1145/344779.344859.
- [2] K. Anjyo and H. Ochiai, *Mathematical basics of motion and deformation in computer graphics*, Synthesis Lectures on Computer Graphics and Animation 6, 3, 1–83, 2014.
- [3] M. Gleicher and A. Witkin, *Through-the-lens camera control*, SIGGRAPH Comput. Graph. 26, 2 (July 1992), 331–340. DOI=10.1145/142920.134088.
- [4] J. E. Humphreys, *Introduction to Lie Algebras and Representation Theory*, Second printing, revised. Graduate Texts in Mathematics, 9. Springer-Verlag, New York, 1978.
- [5] T. Igarashi, T. Moscovich, and J. F. Hughes, *As-rigid-as-possible shape manipulation*, In ACM SIGGRAPH 2005 Papers (SIGGRAPH '05), Markus Gross (Ed.). ACM, New York, NY, USA, 1134–1141. DOI=10.1145/1186822.1073323.
- [6] J. Jeffers, *Lost Theorems of Geometry*, The American Mathematical Monthly Vol. 107, No. 9, pp. 800–812, 2000.
- [7] S. Kaji and G. Liu, *Probe-type deformers*, Mathematical Progress in Expressive Image Synthesis II, Springer-Japan, 63–77, 2015.
- [8] L. Kavan, S. Collins, J. Zara, C. O'Sullivan. *Geometric Skinning with Approximate Dual Quaternion Blending*. ACM Transaction on Graphics, 2008, 27(4), 105.
- [9] H. Ochiai, and K. Anjyo, *Mathematical Description of Motion and Deformation*, SIGGRAPH Asia 2013 Course, <http://portal.acm.org>, 2013. (Revised course notes are also available at <http://mcg.imi.kyushu-u.ac.jp/english/index.php>)
- [10] S. Pinheiro, J. Gomes, and L. Velho, *Interactive Specification of 3D Displacement Vectors Using Arcball*, In Proceedings of the International Conference on Computer Graphics (CGI '99). IEEE Computer Society, Washington, DC, USA, 70.
- [11] S. Schaefer, T. McPhail, and J. Warren, *Image deformation using moving least squares*, In ACM SIGGRAPH 2006 Papers (SIGGRAPH '06). ACM, New York, NY, USA, 533–540. DOI=10.1145/1179352.1141920.
- [12] K. Shoemake, *Animating rotation with quaternion curves*, ACM SIGGRAPH, pp. 245–254, 1985.
- [13] <https://github.com/KyushuUniversityMathematics/iPad-ProbeDeformer>

KYUSHU UNIVERSITY / JST CREST

E-mail address: ma212041@math.kyushu-u.ac.jp

YAMAGUCHI UNIVERSITY / JST CREST

E-mail address, Corresponding author: skaji@yamaguchi-u.ac.jp

KYUSHU UNIVERSITY / JST CREST

E-mail address: ochiai@imi.kyushu-u.ac.jp